

CS210 Data Structures (242) Final Exam

Name: _____ ID _____

Check your section:

<input type="checkbox"/> Dr. Basit Qureshi <input type="checkbox"/> Dr. Syed Umar Amin <input type="checkbox"/> Dr. Sadique Ahmad <input type="checkbox"/> Dr. Adel Ammar	<input type="checkbox"/> Dr. Sawsan Halawani <input type="checkbox"/> Dr. Najla Althuniyan <input type="checkbox"/> Dr. Nada Hakami
--	---

Check you section time

9 am	10 am	11 am	1 pm	2 pm
------	-------	-------	------	------

Instructions:

- This exam contains four questions with multiple parts, on 8 sheets of papers (2 sided).
- Time allowed: 180 minutes
- Closed Book, Closed Notes.
- Use of Calculators is ALLOWED. Use of other computing devices / smartphones etc is strictly prohibited.
- Answer the problems on the exam sheets only. No additional attachments would be accepted.

Question No.	Part a	Part b	Student's Score
Question 1 (CLO 1)	/12	/8	/20
Question 2 (CLO 2)	/5		/5
Question 3 (CLO 3)	/9		/9p
Question 4 (CLO 4)	/6		/6
Total			/40

 Department of
Computer Science

جامعة الامير سلطان
PRINCE SULTAN
UNIVERSITY



CCIS

كلية علوم الحاسب والمعلومات
COLLEGE OF COMPUTER &
INFORMATION SCIENCES

Prince Sultan University
CCIS - Department of Computer Science

<This space is left blank intentionally>

Write your Responses on this sheet only.

Question 1: Part A

- 1. (A) (B) (C) (D)
- 4. (A) (B) (C) (D)
- 7. (A) (B) (C) (D)
- 10. (A) (B) (C) (D)
- 13. (A) (B) (C) (D)
- 16. (A) (B) (C) (D)
- 19. (A) (B) (C) (D)
- 22. (A) (B) (C) (D)

- 2. (A) (B) (C) (D)
- 5. (A) (B) (C) (D)
- 8. (A) (B) (C) (D)
- 11. (A) (B) (C) (D)
- 14. (A) (B) (C) (D)
- 17. (A) (B) (C) (D)
- 20. (A) (B) (C) (D)
- 23. (A) (B) (C) (D)

- 3. (A) (B) (C) (D)
- 6. (A) (B) (C) (D)
- 9. (A) (B) (C) (D)
- 12. (A) (B) (C) (D)
- 15. (A) (B) (C) (D)
- 18. (A) (B) (C) (D)
- 21. (A) (B) (C) (D)
- 24. (A) (B) (C) (D)

Question 1: Part B

- 25. (A) (B) (C) (D)
- 28. (A) (B) (C) (D)
- 31. (A) (B) (C) (D)

- 26. (A) (B) (C) (D)
- 29. (A) (B) (C) (D)
- 32. (A) (B) (C) (D)

- 27. (A) (B) (C) (D)
- 30. (A) (B) (C) (D)

Question 2

- 1. (A) (B) (C) (D)
- 4. (A) (B) (C) (D)
- 7. (A) (B) (C) (D)
- 10. (A) (B) (C) (D)

- 2. (A) (B) (C) (D)
- 5. (A) (B) (C) (D)
- 8. (A) (B) (C) (D)

- 3. (A) (B) (C) (D)
- 6. (A) (B) (C) (D)
- 9. (A) (B) (C) (D)

Question 3

- 1. (A) (B) (C) (D)
- 4. (A) (B) (C) (D)
- 7. (A) (B) (C) (D)

- 2. (A) (B) (C) (D)
- 5. (A) (B) (C) (D)
- 8. (A) (B) (C) (D)

- 3. (A) (B) (C) (D)
- 6. (A) (B) (C) (D)
- 9. (A) (B) (C) (D)

Question 4

- 1. (A) (B) (C) (D)
- 4. (A) (B) (C) (D)

- 2. (A) (B) (C) (D)
- 5. (A) (B) (C) (D)

- 3. (A) (B) (C) (D)
- 6. (A) (B) (C) (D)

<This space is left blank intentionally>

Question 1 [CLO 1]

[/ 20 Points]

Part A:

(/ 12 points)

Answer the following MCQs. Each MCQ is worth 0.5 point:

1. What is the time complexity of inserting a node at the beginning of a singly linked list?			
A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(n^2)$
2. A singly linked list contains n number of names. Searching and removing "Ahmed" from the list would take this time:			
A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(n^2)$
3. Assume that the singly linked list is ordered (sorted in ascending order). In what time can we search a node.			
A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(n^2)$
4. Assume we use a doubly linked list that is ordered. Inserting a node just before the last node would take:			
A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(n^2)$
5. In a singly circular list starting at head, if we start traversing using an iterator, how can we stop?			
A) When iterator = null	B) When iterator = length	C) When iterator = head	D) When iterator = previous
6. In a circular linked list starting at head, how much time would it take to insert a new node before the head.			
A) $O(1)$	B) $O(\log n)$	C) $O(n^2)$	D) $O(n)$
7. A circular queue is implemented with the following structure:			
<ul style="list-style-type: none"> • The queue uses a front and rear index. • Initially, front = -1 and rear = -1, indicating an empty queue. • On enqueue, rear = (rear + 1) % size, and the new element is placed at queue[rear]. • On dequeue, front = (front + 1) % size. 			
Which condition correctly detects overflow (full queue) in the circular queue of size n?			
A) rear == n - 1	B) front == rear	C) (rear + 1) % n == front	D) front == -1
8. In a 0-based array representation of a binary tree, what is the index/position of the parent of the node found at index/position 10?			
A) 5	B) 4	C) 3	D) 2
9. What are the minimum and maximum possible heights of a binary tree with 100 nodes, assume that the root is at height 0?			
A) Min = 7, Max = 100	B) Min = 7, Max = 50	C) Min = 6, Max = 6	D) Min = 6, Max = 99

Prince Sultan University
CCIS - Department of Computer Science

10. The Postorder traversal of a given binary search tree T is: 5, 17, 19, 18, 16, 70, 85, 60, 20,

Which one of the following BSTs represents this T:

<p>A)</p> <pre> graph TD 18 --> 16 18 --> 60 16 --> 5 16 --> 17 60 --> 19 60 --> 70 19 --> 20 19 --> 85 </pre>	<p>B)</p> <pre> graph TD 20 --> 17 20 --> 70 17 --> 16 17 --> 18 16 --> 5 18 --> 19 70 --> 60 70 --> 85 </pre>	<p>C)</p> <pre> graph TD 20 --> 16 20 --> 60 16 --> 5 16 --> 18 18 --> 17 18 --> 19 60 --> 85 </pre>	<p>D)</p> <p>None</p>
--	--	---	-----------------------

11. What is the time complexity of inserting n nodes one by one into an empty AVL tree?

A) $O(n \log n)$	B) $O(n^2)$	C) $O(n)$	D) $O(\log n)$
------------------	-------------	-----------	----------------

12. Suppose you are searching for a key that does not exist in a Binary Search Tree (BST) with n nodes, how many comparisons are performed in the worst case?

A) $O(\log n)$	B) $O(1)$	C) $O(n \log n)$	D) $O(n)$
----------------	-----------	------------------	-----------

13. Suppose you have a complete binary tree. What is the worst-case time complexity for searching a node?

A) $O(\log n)$	B) $O(n)$	C) $O(n \log n)$	D) $O(h)$, where h is the height of the tree.
----------------	-----------	------------------	--

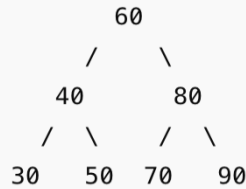
14. What will be the result of inserting a key that already exists in the BST?

A) The key will be inserted again.	B) The method will throw an exception due to the recursion depth.	C) The tree remains unchanged.	D) The method will replace the existing node with the new key.
------------------------------------	---	--------------------------------	--

<This space is left blank intentionally>

Prince Sultan University
CCIS - Department of Computer Science

15. Given the following BST, which of the following trees could be a valid resulting BST after deleting node 60?



<p>A)</p> <pre> 50 / \ 40 80 / \ / \ 30 70 70 90 </pre>	<p>B)</p> <pre> 80 / \ 40 90 / \ / 30 50 70 </pre>	<p>C)</p> <pre> 70 / \ 40 80 / \ \ 30 50 90 </pre>	<p>D)</p> <p>Both A and C are valid</p>
---	--	---	---

16. What is the best-case time complexity for inserting a value into a Binary Search Tree (BST)?

A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(h)$, where h is the height of the tree.
-----------	----------------	-----------	--

17. The worst-case runtime for inserting an edge in a adjacency matrix is:

A) $O(E)$	B) $O(V)$	C) $O(1)$	D) $O(E+V)$
-----------	-----------	-----------	-------------

18. A graph is represented in an edge-list data-structure and has the following edges.

(1,2) (1,3) (1,4) (1,5) (1,6) (1, n)

What would be the worst-case time for searching an edge in this graph?

A) $O(E)$	B) $O(V)$	C) $O(1)$	D) $O(E+V)$
-----------	-----------	-----------	-------------

19. A directed graph is dense (i.e. a lots of edges), which data-structure gives the best space complexity

A) Edge-list	B) Adjacency-list	C) Adjacency-matrix	D) None of these
--------------	-------------------	---------------------	------------------

20. In selection sort, the number of swaps on an array of length n is:

A) $n^2 - n$	B) $n - 1$	C) n	D) $\lfloor n/2 \rfloor$
--------------	------------	--------	--------------------------

21. A quick sort algorithm is designed to pick the first element in an array as a pivot, which of the following arrays gives the worst-case runtime?

A) Items are randomly arranged in the array	B) Items are already sorted in ascending order in the array	C) Items are already sorted in descending order in the array	D) Both B and C
---	---	--	-----------------

Prince Sultan University
CCIS - Department of Computer Science

22. During the merge step of merge sort on two subarrays (total length n), the time complexity is:			
A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(n^2)$
23. Given the array [3, 5, 2, 1, 4], what would be output of the insertion sort after the second pass (i=2)?			
A) [2, 3, 1, 4, 5]	B) [2, 3, 1, 4, 5]	C) [2, 3, 1, 4, 5]	D) [2, 3, 1, 4, 5]
24. Suppose you are merging two sorted subarrays: [4, 6] and [1, 3] using the merge method. Which element(s) remains uncomparred (i.e., not compared with any element from the other subarray during the merge process)?			
A) 3	B) 6	C) 1	D) None

Part B:

(/ 8 points)

Answer the following MCQs. Each MCQ is worth 1 point:

25. Given the sequence [7, 2, 5, 3, 9, 1], inserted one by one into a min-heap, what is the final array representation (Assume a 0-based array implementation, i.e., root is at index 0)?			
A) [1, 2, 5, 3, 9, 7]	B) [1, 3, 2, 7, 9, 5]	C) [1, 2, 3, 5, 7, 9]	D) [1, 3, 2, 5, 7, 9]
26. You are given the following array representation of a binary tree: [20, 18, 15, 10, 12, 9, 8, 5, 7] What property does this array satisfy?			
A) Max-Heap	B) Min-Heap	C) Binary Search Tree	D) AVL
27. The preorder traversal sequence of a binary search tree BST is: 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the postorder traversal sequence of the same tree?			
A) 10, 20, 15, 23, 25, 35, 42, 39, 30	B) 15, 10, 25, 23, 20, 42, 35, 39, 30	C) 15, 10, 23, 25, 20, 35, 42, 39, 30	D) 15, 20, 10, 23, 25, 42, 35, 39, 30

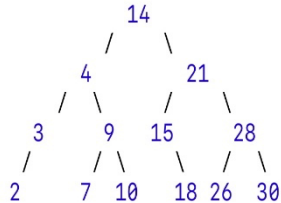
<This space is left blank intentionally>

Prince Sultan University
CCIS - Department of Computer Science

28. Given an empty AVL tree, how many rotations are required when inserting the keys 21, 26, 30, 9, 4 in this exact order to maintain it as a AVL balanced tree?

- A) 0 B) 1 C) 2 D) 3

29. What will be the resulting AVL tree if we delete 2, 3, and 10 in the given order from the following AVL tree?



- | | | | |
|--|---|---|---|
| <p>A)</p> <pre> 14 / \ 9 21 / \ / \ 7 - 15 28 / \ / \ 4 18 26 30 </pre> | <p>B)</p> <pre> 14 / \ 7 21 / \ / \ 4 9 15 28 \ / \ 18 26 30 </pre> | <p>C)</p> <pre> 21 / \ 14 28 / \ / \ 7 15 26 30 / \ \ 4 9 18 </pre> | <p>D)</p> <pre> 14 / \ 4 21 \ / \ 7 15 28 / \ / \ 9 18 26 30 </pre> |
|--|---|---|---|

30. An undirected Graph is stored in a Adjacency Matrix given below. Which of the following gives a correct Depth-First-Search starting at 0. Assume when choosing the next vertex, always go for the lowest value.

```

int[][] adjMatrix = {
    //0 1 2 3 4 5 6
    {0,1,0,1,0,0,0}, // 0
    {1,0,1,0,1,0,0}, // 1
    {0,1,0,0,0,1,0}, // 2
    {1,0,0,0,1,0,0}, // 3
    {0,1,0,1,0,0,0}, // 4
    {0,0,1,0,0,0,1}, // 5
    {0,0,0,0,0,1,0}  // 6
};
  
```

- A) 0, 1, 2, 5, 6, 4, 3 B) 0, 3, 4, 1, 2, 6, 5 C) 0, 1, 2, 4, 3, 5, 6 D) 0, 1, 4, 2, 5, 3, 6

31. For the undirected graph in the previous question, Give the Breadth-First-Search run starting at 0.

- A) 0, 1, 2, 3, 6, 4, 5 B) 0, 3, 4, 1, 2, 6, 5 C) 0, 1, 3, 2, 4, 5, 6 D) 0, 1, 4, 2, 5, 3, 6

Prince Sultan University
CCIS - Department of Computer Science

32. A directed graph is stored in an Adjacency List given below. Which of the following gives a correct Depth-First-Search starting at 0. Assume when choosing the next vertex, always go for the Highest value.

0: 1, 2, 3
1: 2, 5
2: 5
3: 4
4: 6
5: 6, 7
6: 2
7: -

- | | | | |
|------------------------|------------------------|------------------------|------------------------|
| A) 0, 1, 2, 5, 6, 4, 3 | B) 0, 3, 4, 6, 2, 5, 7 | C) 0, 1, 2, 4, 3, 5, 6 | D) 0, 1, 4, 2, 5, 3, 6 |
|------------------------|------------------------|------------------------|------------------------|

<This space is left blank intentionally>

Question 2 [CLO 2]

[/ 5 Points]

Answer the following MCQs. Each MCQ is worth 0.5 point:

1. Hash table T of size 10 uses the hash function $h(x) = x \% 10$. Insert the values: 12, 22, 32, 42. Assuming Linear Probing, how many total collisions will occur?

- | | | | |
|------|------|------|------|
| A) 0 | B) 2 | C) 3 | D) 4 |
|------|------|------|------|

2. A hash table of size 11 uses the hash function $h(x) = x \% 11$. If values 11, 22, 33, 44, 55 are inserted using Chaining, what will be the length of the chain at index 0?

- | | | | |
|------|------|------|------|
| A) 5 | B) 4 | C) 1 | D) 0 |
|------|------|------|------|

3. Using Chaining, insert the keys 17, 29, 31, 42 into a table of size 6 using $h(x) = x \% 6$. Which of the following chains is correct?

- | | | | |
|----------------------|------------------|------------------|----------------------|
| A) Index 5: [29, 31] | B) Index 1: [17] | C) Index 0: [42] | D) Index 1: [17, 29] |
|----------------------|------------------|------------------|----------------------|

Prince Sultan University
CCIS - Department of Computer Science

4. Double Hashing with $h_1(x) = x \% 10$ and $h_2(x) = 7 - (x \% 7)$. Insert 29, 39, 49 in a table of size 10 using Double Hashing. What will be the index of 49?			
A) Index 0	B) Index 1	C) Index 4	D) Index 6
5. If 18, 28, and 38 are inserted into a table of size 10 using linear probing and $h(x) = x \% 10$, then 38 will be placed at index?.			
A) Index 0	B) Index 1	C) Index 8	D) Index 9
6. A hash table of size 7 uses $h(x) = x \% 7$. If values 14, 21, 28 are inserted using Linear Probing, which index does 28 go to?			
A) 0	B) 1	C) 3	D) 2
7. What happens in linear probing when the load factor becomes close to 1 (i.e., the hash table is nearly full)?			
A) Collisions increase	B) Collisions decrease	C) Search-time remains the same	D) memory increases
8. What is the worst-case time complexity of searching for an element in a hash table of size n using linear probing?			
A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(n \log n)$
9. In a hash table of size n, using linear probing assuming uniform distribution of keys, what is the average-case time complexity for inserting a key?			
A) $O(1)$	B) $O(n)$	C) $O(\log n)$	D) $O(n^2)$
10. What is the best-case of time complexity for deleting a key using linear probing?			
A) $O(1)$	B) $O(\log n)$	C) $O(n)$	D) $O(n \log n)$

<This space is left blank intentionally>

Question 3 [CLO 3]

[/ 9 Points]

Give the runtime complexity for each of the following code snippets: Each MCQ is worth 1 point:

<p>1.</p> <pre>public static String M1(String str) { char[] chars = str.toCharArray(); int left = 0, right = chars.length - 1; while (left < right) { char temp = chars[left]; chars[left] = chars[right]; chars[right] = temp; left++; right--; } return new String(chars); }</pre>			
A) O(1)	B) O(logn)	C) O(n)	D) O(n ²)
<p>2.</p> <pre>public static void M2(int[] arr) { for (int i = 0; i < arr.length - 1; i++) { for (int j = 0; j < arr.length - i - 1; j++) { if (arr[j] > arr[j + 1]) { int temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = temp; } } } }</pre>			
A) O(1)	B) O(logn)	C) O(n)	D) O(n ²)
<p>3. public static int M3(int n) {</p> <pre> if (n <= 1) return n; int a = 0, b = 1, c; for (int i = 2; i <= 10; i++) { c = a + b; a = b; b = c; } return b; }</pre>			
A) O(1)	B) O(logn)	C) O(n)	D) O(n ²)

<This space is left blank intentionally>

Prince Sultan University
CCIS - Department of Computer Science

```
4. public static int[][] M4(int[][] a, int[][] b) {
    int m = a.length, n = b[0].length;
    int[][] result = new int[m][n];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            for (int k = 0; k < b.length; k++) {
                result[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    return result;
}
```

A) $O(1)$

B) $O(n)$

C) $O(n^2)$

D) $O(n^3)$

```
5. public static boolean searchAVL(Node itr, int target) {
    if (itr == null) return false;
    if (itr.key == target) {
        return true;
    } else if (target < itr.key) {
        return searchAVL (itr.left, target);
    } else {
        return searchAVL (itr.right, target);
    }
}
```

A) $O(1)$

B) $O(\log n)$

C) (n)

D) $O(n \log n)$

```
6. public static void SortMe(int[] arr) {
    int n = arr.length;
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;

        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}
```

A) $O(1)$

B) $O(\log n)$

C) (n)

D) $O(n^2)$

<This space is left blank intentionally>

Prince Sultan University
CCIS - Department of Computer Science

<pre> 7. public static void H(int n, char from, char to, char aux) { if (n == 1) { System.out.println("1 from " + from + " to " + to); return; } H(n - 1, from, aux, to); System.out.println(n + " from " + from + " to " + to); H(n - 1, aux, to, from); } //Call H(3, "a", "b", "c") </pre>			
A) O (n)	B) O (n log n)	C) (n ²)	D) O(2 ⁿ)
<pre> 8. public static void TimeRecursion(int n) { if (n <= 1) return; System.out.println("Current n: " + n); TimeRecursion (n / 2); } public static void M8(int n) { while(n >= 0) TimeRecursion (n--); } //Call M8(8); </pre>			
A) O (n)	B) O (n log n)	C) (n ²)	D) O(2 ⁿ)
<pre> 9. public int M9(int[] B, int x) { if (x < 0) return 0; else return B[x] + M9(B, x - 2); } M9(new int[]{1,2,3,4,5,6,7,8,9,10}, 9); //Method Call </pre>			
A) O (1)	B) O (log n)	C) (n)	D) O(n log n)

<This space is left blank intentionally>

Question 4 [CLO 4]

[/ 6 Points]

Answer the following MCQs. Each MCQ is worth 1 point:

<p>1. What is wrong with the following code for a removing node <code>third</code> from doubly circular linked list of size 2:</p> <pre>Line1: Node third = cursor.next; Line2: cursor.next = third.next; Line3: third.next.prev = third;</pre>			
A) Line3 should be <code>cursor.next.prev = cursor;</code>	B) cursor becomes <code>null</code>	C) Line3 should be <code>cursor.prev = third;</code>	D) Line3 should be <code>cursor.prev = cursor;</code>
<p>2. You are given the following sequence of operations applied to a stack and a queue simultaneously, both start empty. <code>add(x)</code> inserts the element <code>x</code> into the stack and the queue. <code>remove()</code> removes a single element from both stack and the queue. Which of the following correctly shows the values remaining in each data structure?</p> <p><code>add(1); add(2); add(3); remove(); add(4); remove();</code></p>			
A) Stack: 3, 4 Queue: 1, 2	B) Stack: 3, 2 Queue: 1, 2	C) Stack: 3, 4 Queue: 2, 3	D) Stack: 2, 1 Queue: 3, 4
<p>3. You are given an initially empty priority queue (<code>pq</code>, which is min-heap) and an empty stack. The following operations are performed. What will be the printed output from the above code snippet?</p> <pre>pq.insert(7); pq.insert(3); pq.insert(5); stack.push(pq.remove()); stack.push(pq.remove()); pq.insert(2); stack.push(pq.remove()); while (!stack.isEmpty()) { System.out.print(stack.pop() + " "); }</pre>			
A) 2 5 3	B) 3 5 2	C) 5 3 2	D) 2 3 5

Prince Sultan University
CCIS - Department of Computer Science

4. You are asked to reverse a queue using a stack. Which of the following Java code snippets correctly implements this logic:

<p>A)</p> <pre>Stack<Integer> stack = new Stack<>(); while (!queue.isEmpty()) { queue.dequeue(); } while (!stack.isEmpty()) { queue.enqueue(stack.pop()); }</pre>	<p>B)</p> <pre>Stack<Integer> stack = new Stack<>(); for (int i = 0; i < queue.size(); i++) { stack.push(queue.front()); } while (!stack.isEmpty()) { queue.enqueue(stack.pop()); }</pre>
<p>C)</p> <pre>Stack<Integer> stack = new Stack<>(); while (!queue.isEmpty()) { stack.push(queue.dequeue()); } while (!stack.isEmpty()) { queue.enqueue(stack.pop()); }</pre>	<p>D)</p> <pre>for (int i = 0; i < queue.size(); i++) { queue.enqueue(queue.dequeue()); }</pre>

5. What does the following method m() do on a binary tree, assuming the initial call is made with index= 0 and number_nodes equals the total number of nodes in the tree?

```
boolean m(Node root, int index, int number_nodes) {
    if (root == null)
        return true;
    if (index >= number_nodes)
        return false;
    return (m(root.left, 2 * index + 1, number_nodes)
        && m(root.right, 2 * index + 2, number_nodes));
}
```

A) Checks whether the binary tree is a Binary search tree.	B) Checks whether the binary tree is a complete tree.	C) Checks whether the binary tree is a BST.	D) Checks whether the binary tree is an AVL tree.
--	---	---	---

6. In the following delete method for a BST, which of the following correctly fills in the blanks (a), (b), and (c) to handle deletion when the node to be deleted has two children?

```
Node delete(Node root, int key) {
    if (root == null) return null;
    if (key < root.key)
        root.left = delete(root.left, key);
    else if (key > root.key)
        root.right = delete(root.right, key);
    else {
        if (root.left == null) return root.right;
        if (root.right == null) return root.left;

        Node n= findMinNode(root.right);
        root.key = _____ (a)
        _____ (b) = delete(_____, _____) (c);
    }
    return root;
}
```

A) (a) n.key (b) root.right (c) root.right, n.key	B) (a) n.right.key (b) root (c) root, n.right.key	C) (a) n.left.key (b) root.left (c) root.left, n.key	D) (a) n.key (b) root.left (c) root.left, n.key
--	--	---	--

-END OF EXAM-